CMSC 23700 **Introduction to Computer Graphics** **Project 1a**
Fall 2005 **October 19**

Ray tracer (Part a)

Due: October 28

As with Project 0, we will create a project on the gforge server. This project will be seeded with code that implements the GML compiler and interpreter. In addition, we have provided partial implementations of some of the GML operations. For Part-a of the project, you are responsible for the following GML operations:

Name	Description
`light`	defines a directional light source
`plane`	the XZ-plane
`pointlight`	defines a point-light source
`render`	render a scene to a file
`rotatex`	rotation around the X-axis
`rotatey`	rotation around the Y-axis
`rotatez`	rotation around the Z-axis
`scale`	scaling transform
`sphere`	a unit sphere
`translate`	translation transform
`union`	union of two solids
`uscale`	uniform scaling transform

See Handout 3 (Project 1 Overview) for details on these operations.

The sample code provides implementations of the GML operations for lights, shapes, and CSG operators. This code, however, just deals with the allocation and initialization of the representations of these objects; you are responsible for writing the intersection testing code.

Note that while you do not have to implement the CSG intersection and difference operations in this stage, you should be mindful of their existence.

We will collect the projects at 9pm on Friday October 28th from the repositories, so make sure that you have committed your final version before then.

The Makefile in the repository builds an executable program called `gml`. To raytrace a given GML file, we call this program with the filename as a command-line argument. For example, assume we have a GML file `scene.gml`, then the following command raytrace the scene specified in the file:

```
% gml scene.gml
```